Size to fit browser

Patent number:	AU761635
Publication date:	2003-06-05
Inventor:	KELSEY JULIAN BENJAMIN; REEVES JOHN STEWART
Applicant:	CANON KK
Classification:	
- international:	G06F17/00; G09G5/00
- european:	
Application number:	AU20000071514 20001109
Priority number(s):	AU20000071514 20001109; AU1999PQ04034 19991112

Report a data error here

Abstract not available for AU761635

Data supplied from the ***esp@cenet*** database - Worldwide

S&F Ref: 529280

AUSTRALIA

PATENTS ACT 1990

COMPLETE SPECIFICATION

FOR A STANDARD PATENT

ORIGINAL

Name and Address of Applicant :	Canon Kabushiki Kaisha 30-2, Shimomaruko 3-chome, Ohta-ku Tokyo 146 Japan
Actual Inventor(s):	Julian Benjamin Kelsey and John Stewart Reeves
Address for Service:	Spruson & Ferguson St Martins Tower 31 Market Street Sydney NSW 2000
Invention Title:	Size to Fit Browser

ASSOCIATED PROVISIONAL APPLICATION DETAILS

[33] Country	[31] Applic. No(s)	[32] Application Date
AU	PQ4034	12 Nov 1999

The following statement is a full description of this invention, including the best method of performing it known to me/us:-

SIZE TO FIT BROWSER

Field of the Invention

The present invention relates to the viewing of electronic documents and, in particular, to a method by which documents, irrespective of their size and in particular their width, are conveniently displayed to enable quick review by a user of the electronic document.

Background

The concept of electronically previewing documents is one well known in the art. For example, many word processor applications and Internet web browser applications include arrangements by which the main editing or viewing display may be represented to the user so as to depict the full layout of a page that may be printed, thus giving rise to the common name "print preview".

However the print preview function of many computer applications is determined by the page size upon which the electronic information is to be printed and a level of magnification desired by the user for the print preview. For example, Microsoft Word for Windows 97 provides a print preview function which can accommodate a number of different paper sizes (eg. A4, foolscap, etc.) and may also reproduce a print preview and a variety of magnifications having preset values from 10% through to 500%. However, with a typical implementation of Word 97, only those print previews having a magnification of 40% or less are sufficient to provide an unobscured full page representation of a single A4 page for example.

Electronic pages developed for presentation on the World Wide Web (WWW or simply "the Web") using Hypertext Mark-up Language (HTML) protocol have no concept of width. The author of such a page may develop the page to any desired width. In order to present an HTML page, Web browser application software basically implement a laying out of the page from the top left hand corner of the display screen, and continue from that "origin" position. The overall width of an HTML page can be determined by those images, tables or frames contained within the page, in combination with various layout rules implemented by the Web browser application.

Web browser applications such as NETSCAPE NAVIGATOR (Netscape Corporation, USA), and MICROSOFT INTERNET EXPLORER (Microsoft Corporation, USA) are designed primarily for displaying page contents on a display screen, having printing of the page being relegated to a secondary consideration.

It is an object of the present invention to substantially overcome, or at least ameliorate, one or more deficiencies associated with the existing arrangement.

Summary of the Invention

According to an aspect of the present disclosure, there is provided a method of presenting information comprising plural components by electronic display, said components being collectively arranged for presentation, said method comprising the steps of:

establishing a presentation window having a width;

determining those first ones of said components that contribute to a width of said presentation;

adjusting display sizes of said first components by applying a plurality of differing scaling factors to widths of said first components, so that said width of said presentation is adjusted to fall within said width of said presentation window while maintaining a layout corresponding to said presentation; and

displaying the adjusted information within said presentation window.

Other aspects are also disclosed.

Brief Description of the Drawings

A number of embodiments of the present invention will now be described with reference to the drawings and Appendix, in which:

Fig. 1 illustrates a Web browser page presentation according to the prior art;

Fig. 2 is another representation of a prior art Web browser representation;

Fig. 3 is a presentation of a size to fit browser in accordance with the present disclosure;

Fig. 4 is a further presentation in accordance with the present disclosure;

Figs. 5A and 5B provide a first comparative example;

Figs. 6A and 6B provide a second comparative example;

Figs. 7A and 7B provide further examples according to the present disclosure;

Figs. 8A and 8B are prior art and Fig. 8C is a comparative example of a size to fit widow;

Fig. 9 is a schematic block diagram representation of a computer system in which embodiments of the invention may be practised;

Figs. 10A to 10D depict one method of size-to-fit layout;

Figs. 11A and 11B are flow charts of a size-to-fit layout method;

Figs. 12A and 12B are graphs depicting relationships between page components and their layout according to an alternative implementation; and

Appendix A provides a pseudo-code description of a size-to-fit layout method including differential scaling.

Detailed Description

Fig. 1 shows a representation of a display window 10 of a prior art Web browser display formed using the NETSCAPE browser application. The window 10 may be resized in a known manner utilising a resizing corner tab 11 or by dragging a side boundary 12 and/or 13. Having established the size of the window 10, information contained at a particular Web page may then be presented (displayed) within a display area 14 of the window 10. In this example, it is seen that the information displayed within the display area 14 exceeds the boundaries of the display area 14 and, as a consequence, a vertical scroll bar 15 and a horizontal scroll bar 16 are provided to allow for two-dimensional scrolling of the display area 14 so that all of the contents of the Web page may be viewed.

Fig. 2 shows a window 20 that corresponds to the window 10 in all respects excepting that the window 20 has been resized, for example through a manipulation of the corner tab 11. Such resizing will be well known to those familiar with operating systems that utilize windows to present information. It will be observed from Fig. 2 that although the same information contained by the window 10 is presented in the window 20, the manner in which that information is displayed has been altered. In particular, in Fig. 1, a first portion 17 of the information is displayed spread across the display area 14 whereas in Fig. 2, the corresponding information 22 has been repositioned in an attempt to allow that information to fit within the window 20.

Significantly, with each of the prior art arrangements of Figs. 1 and 2 a horizontal scroll bar is necessary to enable viewing of all the contents of the Web page. Further, in view of the dependence of the information being displayed upon the size of the browser window, the author of the Web page could never be 100% certain the Web page will be shown as was intended when created.

With each of the presentations of Figs. 1 and 2, the user has the option of selecting a print preview function which operates in the fashion described above to provide a print preview of a page to be printed using the information contained. It will be appreciated by those skilled in the art that the print preview corresponding to each of Figs. 1 and 2 will be identical. However, depending on the authored width of the Web page, some information contained therein may not be represented in the print preview.

Preferred embodiments of the present invention relate to a window dubbed with the name "Quick View" by the present inventors that, when selected, automatically provides a presentation within a window of the full width of the Web page in a fashion automatically sized to fill the width of the window.

Turning now to Fig. 3, a window 30 is shown according to the preferred embodiment derived from the same source information as Figs. 1 and 2. The size of the window 30 is adjustable in a manner corresponding the windows of Figs. 1 and 2. However, and in accordance with the preferred embodiment, the window 30, when selected by the user, provides a presentation 31 of the information contained within the Web page automatically sized to display the entire width of the Web page within the window 30. As will be apparent, the content of the Web page shown in the window 30 of Fig. 3 corresponds to that shown in each of the windows 10 and 20 of Figs. 1 and 2 respectively, but also includes further information contained within the particular Web page that would only be visible in the windows 10 and 20 through manipulation of the scroll bars 15 and 16.

Fig. 4 shows another window 40 in accordance with the preferred embodiment also derived from the same source information as Figs. 1 and 2. In comparison with the presentation of Fig. 3, it will be appreciated that the window 40 has been resized and reshaped which has resulted in a resizing of the corresponding display area 41. However, in similar fashion to the window 30 of Fig. 3, the same Web page information is again presented in a "Quick View" fashion such that the display area 41 contains the entire width of the Web page. As will be appreciated through a comparison of Figs. 3 and 4, and the prior art presentations of Figs. 1 and 2, that all of the information contained within the Web page is again reproduced but, in view of the smaller size of the window 40 compared to the window 30, also the information is reproduced in a much smaller fashion (lower zoom ratio) sized to fit within the width of the window 40.

As seen in each of Figs. 3 and 4, vertical scroll bars 32 and 42 are provided to enable viewing of those Web pages which, when presented in the "Quick View" format, exceed the length of the window. This facility is provided in the described embodiments since users of windowed applications are usually adept at scrolling in up and down directions through a document relatively quickly in order to obtain an impression of the contents thereof. Scrolling however in two-dimensions, as required by the prior art arrangements of Figs. 1 and 2 is significantly more time consuming and can be disorientating depending on the zoom ratio of the document being presented.

Obviating a need for horizontal scrolling in the described embodiments conveniently provides for the display of frames, tables and images that often extend beyond with the width of a normal browser display window (eg. Figs 1 and 2) or a printable page (eg. a print preview display). In this regard, with many Web documents, a traditional "print preview" using a standard browser such as those shown in Figs. 1 and 2,

can still result in some of the information of the Web page being chopped off from the edge of the print preview display.

The preferred embodiment facilitates the placing of two of the "Quick View" windows 30 or 40 side-by-side on a normal size computer screen, so that the contents of each can be read and compared without the need for any horizontal scrolling in either window. For prior art browser displays, if two browser windows are placed side by side, in general, horizontal scrolling will be necessary in order to view arbitrary Web pages.

Although the text of the window 30 is difficult to read, and the smaller text of the window 40 is essentially illegible, such presentations can still be useful in providing the user with a quick summary of the content of a page. In particular, in many cases, text may be sufficiently legible to permit the collecting of links and the previewing of Web sites and the comparison of Web pages.

The "Quick View" windows 30 and 40 further do not correspond with print preview windows which are founded upon a specific height to width ratio determined by the page medium upon which the information is to be printed (eg. A4 or foolscap etc). In the preferred embodiment, the size of the presentation is determined by the content of the electronic Web page and the size of the "Quick View" window into which it is to be presented.

In order to provide the "Quick View" presentation, the embodiments apply a size-to-fit layout method which comprises differential scaling and layout to those distinct components of the Web page to be presented. In this regard, most Web pages include a variety of data formats and types which, by virtue of the HTML structure and other similar structures, divide the Web page into component blocks such as text, tables, images (eg. JPEG, GIF), uniform resource indicators (URI's) and locators (URL's), and the like. The preferred embodiment operates by identifying those different components that form a Web page to be presented, and then applies different scaling factors and layout arrangements to the components in order to optimise their respective display within the "Quick View" window whilst maintaining the general layout of the Web page as established by the author thereof. This can be achieved according to one embodiment by examining those components that contribute to the widest portion of the Web page, and establishing scaling factors for those components to enable them to fit within the "Quick View" window. The scaling factors of other components are then determined so as to maximise the presentation size of those other components whilst maintaining the general layout of the "Quick View" presentation consistent with that of the Web page as

originally authored. Examples of the differential scaling used in the preferred embodiment can now be described with reference to Figs. 5A to 8B.

Fig. 5A shows a prior art NETSCAPE window 50 comprising a number of components including a banner 51A (a JPEG image), a text layout 56A including URI's, and a table 57 which includes a text sub-table 57A incorporating URI's and an image portion 52A, being the same JPEG image as the banner 51A. As seen, not all of the Web page content is displayed in Fig. 5A. Fig. 5B shows a corresponding "Quick View" window 59 from which it is seen that the full extent of the width of the Web page is set by the table 57 which, as is now apparent includes a further banner image 58B, again formed by the same JPEG image as above. In Fig. 5A, each of the JPEG images 51A and 52A have the same size, in this case represented by a height 53 which has a dimension of 38 mm. As seen from Fig. 5B however, the height 54 of the image 51B (measuring 23 mm) is different from the height 55 of the image 52B (which measures 14 mm). In the preferred embodiment, the scaling factors applied to different Web page components are zoom factors which may be expressed as percentages and thus maintain the height-to-width ratio of the components being scaled. As a consequence, the heights referred to in Figs. 5A and 5B are directly proportional to the corresponding widths of the corresponding components. Heights are referred to in Figs. 5A and 5B for illustrative purposes since in Fig. 5A, the width of the images 51A and 52A are unable to be determined. From the heights mentioned, it will be appreciated that in order to form the "Quick View" presentation of Fig. 5B, the table 57 including the text table 57A, image 52A and image 58B, have each been scaled by 36.8% to enable them to fit within the window 59 in a side-by-side relationship as established by the author of the Web page. The image 51A however, since it is not a determinator of the width of the Web page has been scaled by a different, smaller ratio, in this case 60.5%, to ensure optimal relative reproduction in the window 59.

Figs. 6A and 6B show another comparative example that illustrates how modifications to text may occur. Fig. 6A shows a NETSCAPE window 60 having two components 61A and 62A. In this example, the component 62A comprises a table including text, the table having a specified HTML width which establishes the width of the Web page. Those familiar with the NETSCAPE product will appreciate that some wrapping of the text as seen in Fig. 6A has occurred where that application has attempted to display as much of the Web page as possible within the window 60. Specifically the authored HTML Web page as seen in the window 60 provides for the text within each of the components 61A and 62A to have the same font size. In creating the corresponding

"Quick View" window 59 of Fig. 6B, the extent of scaling of the content of the Web page is founded upon the scaling required for the width determining table 62A to fit within the window 59. A scaling of about 33% is applied to create the corresponding component 62B in the window 69. However, since the portion 61A does not contribute to the maximum width of the Web page, the portion 61A may be scaled by a smaller amount, in this case about 37%. Again, the authored layout of the Web page is maintained.

Fig. 7A shows a "Quick View" window 70 where the maximum width of the displayed presentation is determined by a table including an image component 71 and a text table component 72. In this example, the text table component 72 may be displayed with the text being wrapped thereby minimising the width of the text, and thus the width of the component 72 as displayed. Accordingly, the amount by which the text within the component 72 must be scaled is minimised, thereby providing larger text and thus optimising the reproduction thereof. As seen in Fig. 7A, a vertical scroll bar 73 may be used to view the content of the Web page (ie. part of the text component 72) not presented within the window 70.

Fig. 7B shows a "Quick view" window 74 having the same size as the window 70 and same content excepting that the relatively wide image 71 of Fig. 7A has been replaced by a narrower image 75. Again the image 75 and text table component establish the maximum width of the Web Page. However, in this example, the degree of scaling required for the "Quick View" presentation is not as great, and accordingly, the text component may be represented without wrapping as seen at 76 in Fig. 7B.

Fig. 8A shows a prior art window 80 that may be obtained from a Web page using a traditional browser. In this example, the maximum width of the Web page is set by a very wide image 81 (2000 pixels wide) and text 82 shown at actual size. The text 82 is wrapped and thus may be viewed in full by vertical scrolling. It is observed however that a horizontal scroll bar 83 is required to be manipulated for the entire image 81 to be viewed from within the window 80.

Fig. 8B shows a further prior art window 84 having a width corresponding to the window 80 and presenting all the elements of the Web page of Fig. 8A reduced by the same scaling (zoom) factor. This is a feature available in Word97 for Windows (Microsoft Corp.) that may be achieved by previewing pages with the width set to the window width. As seen in Fig. 8B, the entire image 85 is now presented, but that the text 86, having been shrunk by the same amount, is now so small as to be illegible.

Fig. 8C shows the same Web page as Figs. 8A and 8B viewed in a "Quick View" window 87 having the same width as the windows 80 and 84. The presentation within the window 87 is formed using differential scaling where it is seen that the image 88 has been shrunk to fit the width of the window 87, and in this regard corresponds to the image 85 presented in the window 84. However, through application of differential scaling, the text 89 has remained the same size as it would have without the presence of the image 88, and so remains quite legible.

Some Web pages are created as frames pages, where the main page specifies the size, position and URL's for a number of rectangular frames which are fitted together to form a composite page. Browsers that are capable of displaying frames pages typically display the frames as separate scrollable windows tiled into the mains display area of the browser.

The preferred embodiment handles frames by first converting the frames into HTML tables, and then laying the tables out as it would for other ordinary HTML tables. One consequence of this is that no frame scroll bars are displayed. The frames are displayed in a fixed, tiled arrangement the entirety of which may be scrolled vertically, where necessary. Frames are extended by white space at the bottom and to the right as necessary to make them fit into the tiled arrangement.

The "Quick View" display methods described herein are preferably practiced using a conventional general-purpose computer system 100, such as that shown in Fig. 9 wherein the processes of pseudo-code of Appendix A may be implemented as software, such as an application program, executing within the computer system 100. In particular, the steps of the viewing method are effected by instructions in the software that are carried out by the computer. The software may be divided into two separate parts; one part for carrying out the viewing methods; and another part to manage the user interface between the latter and the user. The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer from the computer readable medium, and then executed by the computer. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer preferably effects an advantageous apparatus for Web page viewing in accordance with the embodiments of the invention.

A seen in Fig. 9, the computer system 100 comprises a computer module 101, input devices such as a keyboard 102 and mouse 103, output devices including a printer 115 and a display device 114. The windows described above for both the prior art

of for the embodiments are typically presented to the user via the display device 114. Manipulation of the windows may be obtained using the mouse 103 and/or keyboard 102 in a traditional manner. A Modulator-Demodulator (Modem) transceiver device 116 is used by the computer module 101 for communicating to and from a communications network 120, for example connectable via a telephone line 121 or other functional medium. The modem 116 can be used to obtain access to computer and communications networks such as Local Area Networks (LAN's) or Wide Area Networks (WAN's), the Internet, and the Web.

The computer module 101 typically includes at least one processor unit 105, a memory unit 106, for example formed from semiconductor random access memory (RAM) and read only memory (ROM), input/output (I/O) interfaces including a video interface 107, and an I/O interface 113 for the keyboard 102 and mouse 103 and optionally a joystick (not illustrated), and an interface 108 for the modem 116. A storage device 109 is provided and typically includes a hard disk drive 110 and a floppy disk drive 111. A magnetic tape drive (not illustrated) may also be used. A CD-ROM drive 112 is typically provided as a non-volatile source of data. The components 105 to 113 of the computer module 101, typically communicate via an interconnected bus 104 and in a manner which results in a conventional mode of operation of the computer system 100 known to those in the relevant art. Examples of computers on which the embodiments can be practised include IBM-PC's and compatibles, Sun Sparcstations or alike computer systems evolved therefrom.

Typically, the application program of the preferred embodiment is resident on the hard disk drive 110 and read and controlled in its execution by the processor 105. Intermediate storage of the program and any data fetched from the network 120 may be accomplished using the semiconductor memory 106, possibly in concert with the hard disk drive 110. In some instances, the application program may be supplied to the user encoded on a CD-ROM or floppy disk and read via the corresponding drive 112 or 111, or alternatively may be read by the user from the network 120 via the modem device 116. Still further, the software can also be loaded into the computer system 100 from other computer readable medium including magnetic tape, a ROM or integrated circuit, a magneto-optical disk, a radio or infra-red transmission channel between the computer module 101 and another device, a computer readable card such as a PCMCIA card, and the Internet and Intranets including e-mail transmissions and information recorded on Websites and the like. The foregoing is merely exemplary of relevant computer readable

mediums. Other computer readable mediums may be practiced without departing from the scope and spirit of the invention.

The viewing method may alternatively be implemented in dedicated hardware such as one or more integrated circuits performing the functions or sub functions of Appendix A. Such dedicated hardware may include graphic processors, digital signal processors, or one or more microprocessors and associated memories.

Referring to Appendix A, the opening or creation of a "Quick View" window may be invoked in manner similar to known print preview windows from a host application, such as a browser. In such cases the window will typically open to a predefined size and thereafter may have its size manipulated by the user as described above. The preferred embodiment as set out in Appendix A operates by processing the Web page on a line-by-line basis, noting that each line may include text, images, tables or combinations thereof. The pseudo-code comprises a main routine PAGE LAYOUT which initially sets the layout width to that of the "Quick View" window. Thereafter each component (or element) of the Web page is processed, on a line-by-line basis. A test is performed to determine if the element is either an image, a table, or other (eg. text). Images and tables are processed according to corresponding subroutines IMAGE LAYOUT and TABLE LAYOUT. The processing of images determines an appropriate scaling factor for that image or enclosing table cell where applicable, and then scales the image accordingly. For tables, elements within the table are considered to determine if they are also images, tables or other. The processing of tables modifies the width of cells in the table to accommodate the elements therewithin. Text, as seen in Fig. 7A may be wrapped within cells whilst images may not. Where an image and a table are present on the same line in the original HTML source, the preferred "Quick View" application may split the two onto separate lines if they don't both fit onto the same line. If the Web page author wishes to prevent such "post production" formatting, the author must create a table and place the components into adjacent table cells.

The processing afforded by an application program according to Appendix A results in the creation of a "Quick View" window of the form described above. Where the user desires to interactively modify the size of the "Quick View" window, manipulation of a resizing corner or edges of the "Quick View" window may be performed in a traditional manner. The "Quick View" presentation within the window is then simply scaled according to the prior art method of Fig. 8B since, at that stage, the "Quick View" presentation is optimised to the width of the window given the particular

source components (elements) from the Web page, and further resizing maintains a constant height to width ratio.

Figs. 10A to 10D show an alternative method for producing a size-to-fit window display. Fig. 10A depicts major components of an HTML coded web page 200 which, when obtained in response to an Internet request includes a certain structure 202, and document content which can include banner images 204, tables 206, isolated images 208 as well as text 210. The present arrangement creates the size-to-fit window as a two-stage process. The first stage requires the creation of a virtual page 220, seen illustrated in Fig. 10B. The virtual page 220 requires as a layout stored electronically within the computer system and becomes a basis from which any size to fit windows are generated. As seen, in Fig. 10B, the virtual page 220 incorporates a banner image 222 together with a table comprising text 226 and image 224 components. The virtual page 220 is arranged according to a predetermined standard display size which does not require the use of a horizontal scroll bar. The virtual page 220 in this fashion may be formed according to the various embodiments previously described. The virtual page 220 however is not directly displayed to the user but rather is displayed in a modified form once the user has established the desired window size. In this regard, the virtual page 220 may be formed so that the components of the documents are sized in an appropriate fashion to facilitate their interpretation, this being particularly important in respect of the text 226.

Once the user has decided on a particular window size within which the HTML code is to be displayed, the virtual page image 220 is then scaled in an appropriate fashion to provide for the display in the appropriate window sizes.

In respect of the window size shown in Fig. 10C, it is seen that the virtual page 220 is scaled by expanding the width and contracting the height of the page. As seen however, the proportional sizes of the images 232 and 234 remain unchanged, although their sizes are scaled in a "percentage" fashion. The text 236 is scaled to occupy the appropriate space available arising from the space vacated according to the size of the banner image 232 and table image 234.

The size to fit window 240 of Fig. 10D may be similarly formed directly from the virtual page 220 of Fig. 10D again through the scaling of individual components of the image.

With the arrangement shown in Figs. 10A to 10D, the two-step arrangement results in the display of a size to fit window and can be employed in such a way that the first step is performed according to various differential scaling procedures disclosed herein and the second step, of resizing the virtual page to fit the desired window size, may

be performed in some cases according to standard scaling procedures known in the art and, for example, depicted in some of the prior art representations contained in this document.

Turning now to Fig. 11A, a method 300 of implementing differential scaling is depicted. The method commences with step 302 where the content source data of the web page is obtained, this being performed by a traditional web access.

At step 304, the layout parameters of the web page are obtained. The layout parameters includes the actual pixel size of the output window and, where appropriate, information regarding the paper upon which that may be output. The parameters also include a notional size of the document in layout units which may be, for example, millimetres, points or inches. A particular device mapping is also provided which is often in dots per inch (dpi). These factors influence the following step 306 which decides on the base scaling factor for the web page. For instance, if the device mapping is ten pixels per millimetre and the notional size is 10cm, the natural width for the window is 1000 pixels.

Step 306 also involves a number of considerations. The simplest scaling factor is to uniformly use the ratio of the natural window width to the actual window width desired by the user. For instance, if the natural window width is 1000 pixels, and the actual width is 500 pixels, then a uniform scaling factor of 50% would allow the output to fit in the actual window without scrolling. This is the solution used by many fit to window page view processes of word processor and like applications. According to the present disclosure different elements have applied to them different scales which may or may not be linearly related to the simple ratio of actual window width to the natural window width. Experiments conducted by the present inventors indicate that images scale on a simple scale effectively, but that text scales best if it is scaled more slowly. This will result in different layouts for any resizing of the window. A graphical example of this is illustrated in Fig. 12A.

An alternative approach can include that illustrated in Fig. 12B. In Fig. 12B, a similar graph has a stepped line for text indicating that font sizes may change in steps (eg. 10 point → 11 point →12 point, etc) where a point is a 1/72 inch, and the window scaling may be more finely grained. The title indicated in Fig. 12B may refer to text in the drawing bar which is very likely to be constant as is indicated.

The base scaling factors decided in step 306 are arbitrary or, to some extent, prevailing scales. Subsequent processing may cause additional scaling of certain components.

Step 308 which follows considers the individual objects of the web page to be laid out. Each object is considered in turn. Step 310 assesses whether or not the next object is a simple object. In this situation, tables are treated as special cases and extra work goes into attending to their optimised layout. Accordingly, simple objects include images and text. Where the object is a table, step 312 follows for specialised processing.

If the object is simple, step 314 follows and a determination is made whether or not the base scaling factor enables the object to fit. If this is not the case, step 316 follows and additional scaling factor is calculated to ensure the object fits. In general, the preferred implementations avoid shrinking text, as has been discussed above. In this regard, a preferred implementation ignores instructions of the author of the Web page which include comments that in effect say "do not wrap" (such as HTML tags "pre" and "nubr"). The preferred implementation wraps text and splits words before any efforts are made to shrink the text. Once the scaling factor is made to work for each object, step 318 generates a render command that is suitably scaled for that object and control returns to step 308 to consider the next object. In respect to step 312, once the table scale and cell widths have been determined, laying out the cell contents is a process of performing the same tasks upon individual components within the cell.

Turning now to Fig. 11B, this shows the specific steps performed within step 312 for handling tables. This process 312 commences with a step 330 where preparations are made in order to lay out a table. This is followed by step 332 which considers each new row in the table in turn. Step 334 gets the information for the first cell in the table row. The cell information may include a maximum width as if laid out on an infinitely wide page. This is a good measure of the amount of content in the cell. The cell information also includes a minimum width. If the minimum width of all the cells in a row is wider than the space available, then the row and the table will need to be scaled. The cell information may also include user hints for size and layout as developed by the author of the web page. Cell widths are specified as pixels and percentages. Percentages are normally part of a given table width overall or else relate to the window width.

A cell may also contain additional tables which are evaluated for min/max widths and author hints in full as part of the processing of the parent cell. Once step 336 is performed for each cell in the row, control follows to step 338 which completes the process for every row of the table through returns to steps 332. Once this is completed, step 340 devises an overall scaling factor and scaling widths for the table. The factors obtained from steps 336 and 338, being the minimum/maximum widths and author hints are known, processing to determine scaling may be needed. The significant goals are to

keep all content legible within the window, to keep all images scaled by a constant factor, and to avoid reducing the size of text. Step 342 follows with a development of the final factors which generate the table layout.

The preferred embodiment as described above lays out electronic information to the width of the window in which the information is to be presented. Images, tables, frames, long words and large controls are laid out to fit given width by avoiding the need to identify the layout of the information such as seen in Figs. 1 and 2 the wrapping text and the like in order to ensure it is displayed. Where any components of the information require shrinking to fit a display area, components of the Web page are shrunk by differing amounts thus providing for single representation within the space available. Another variation may be produced by reducing or increasing the physical width used for the layout in some ratio of the change of the width of the "Quick View" window. Then, upon resizing of the "Quick View" window, part of the size reduction or increase, respectively, may derive from laying out to a smaller or greater, respectively, physical width and the rest may still be obtained by scaling the layout to the size of the window. This may be used to provide for text to shrink "slowly" so as to increase legibility from narrower "Quick View" windows. Other variations any scaling between text and images are also possible.

In a further embodiment, the "Quick View" window need not be presented as a static, "view only" window, but may be active as a browser window for the Web page being presented. In this regard, URI's may be active and selectable, for example through a traditional mouse operation. As such, the "Quick View" window included as a sub-routine to existing Web browser applications and can, to a large extent, replace the browsing use of such applications. This is achieved through a more user friendly presentation of Web page information in a fashion that is conducive to rapid browsing. Where information is identified for which closer examination is required, the user may revert from the "Quick View" window to the traditional browser window in order to resolve any lack of detail contained in the former.

Industrial Applicability

It is apparent from the above that the embodiment(s) of the invention are applicable to the computer and data processing industries, viewing software generally, and Web browser application software in particular.

The foregoing describes only some embodiments of the present invention, and modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the embodiment(s) being illustrative and not restrictive.

In the context of this specification, the word "comprising" means "including principally but not necessarily solely" or "having" or "including" and not "consisting only of". Variations of the word comprising, such as "comprise" and "comprises" have corresponding meanings.

5

Appendix A

Below is a pseudo-code description of the size-to-fit layout method with differential scaling used in the preferred embodiment form the generation of a Quick View window.

```
PAGE LAYOUT
   Set layout fixed width as current Quick View window width
   LOOP for each element to be laid out.
      IF next element is an Image THEN
         DO IMAGE LAYOUT
      ELSE IF next element is a Table THEN
         DO TABLE LAYOUT
      ELSE (eg. Text)
         IF there is room to fit the next element in the current line THEN
            Add the next element to the current line
         ELSE
            Begin a new line.  Add the next element to the new line.
   END LOOP
END PAGE LAYOUT

IMAGE LAYOUT
   Determine Image size:
      IF Image has defined width AND height THEN
         Use these dimensions
      ELSE IF Image has defined width OR height THEN
         Use the defined dimension and calculate the other accordingly
         to maintain the image aspect ratio
      ELSE
         Use the Image natural dimensions.

   Set Image Scale Factor to 1
   IF Image width does NOT fit in current line AND NOT at start of line THEN
      Begin a new line.
   IF Image width does NOT fit in current line THEN
```

```
    Calculate Image Scale Factor = Available width / Image width
  Scale image - multiply image width and height by Image Scale Factor
  Add scaled image to current line.
END IMAGE LAYOUT

TABLE LAYOUT
Calculate Table Minimum Width and Table Maximum Width:
  LOOP for each cell in table
    Calculate element widths for all elements in cell, where elements
      can be words, glyphs, control boxes, images or tables.
      For image elements, determine image size as in IMAGE LAYOUT.
      For table elements, determine table width as Table Maximum Width,
      calculated by performing TABLE LAYOUT recursively on the table
      element.
    Calculate paragraph widths for all paragraphs in cell, by combining
      elements into paragraphs as specified by HTML, and then summing
      element widths to get paragraph widths.
    Calculate Cell Minimum Width = maximum element width
    Calculate Cell Maximum Width = maximum paragraph width
    IF Cell spans multiple columns THEN
      Divide Cell Minimum Width and Cell Maximum Width by the number
      of columns spanned to calculate the widths contributed to each column
      by the cell.
  END LOOP

  LOOP for each column in table
    Calculate Column Minimum Width = maximum Cell Minimum Width for
      all cells in column.
    Calculate Column Maximum Width = maximum Cell Maximum Width for
      all cells in column.
  END LOOP

  Calculate Table Minimum Width = Sum of Column Minimum Width
  Calculate Table Maximum Width = Sum of Column Maximum Width

  IF the Table Maximum Width fits in the current line THEN
    Add table to current line, using Table Maximum Width.
```

```
ELSE
    IF NOT at start of line THEN
        Begin new line
    IF the Table Maximum Width fits in the current (new) line THEN
        Add table to current line, using Table Maximum Width.
    ELSE IF the Table Minimum Width fits in the current line THEN
        Add table to current line, using Table Minimum Width.
    ELSE
        Calculate Table Scale Factor = Available Width / Table Minimum Width
        Scale entire table (ie, all elements in all cells) by Table Scale Factor
        Add scaled table to current line.
END TABLE LAYOUT
```

The claims defining the invention are as follows:

1. A method of presenting information comprising plural components by electronic display, said components being collectively arranged for presentation, said method comprising the steps of:

establishing a presentation window having a width;

determining those first ones of said components that contribute to a width of said presentation;

adjusting display sizes of said first components by applying a plurality of differing scaling factors to widths of said first components, so that said width of said presentation is adjusted to fall within said width of said presentation window while maintaining a layout corresponding to said presentation; and

displaying the adjusted information within said presentation window.

2. A method according to claim 1, wherein said width of at least one of said first components is adjusted by resizing said first component.

3. A method according to claim 1 wherein at least one of said components is a text component, and said adjusting of said display size of said text component comprises wrapping text within said text component to adjust the width of said text component.

4. A method according to claim 1 wherein said first components comprise a primary component that defines said width of said presentation and a plurality of secondary components arranged within said primary component according to said layout.

5. A method according to claim 1 wherein said components are collectively arranged in plural lines of said layout, and said determining comprises examining each said line for first components, and said adjusting comprises applying at least one scaling factor to those first components on said line.

6. A method according to claim 5 wherein said components comprise a primary component that defines a maximum width of said line and a plurality of secondary components arranged within said primary component according to said layout.

7. A method according to claim 4 or 6 wherein said primary component comprises a (main) table and each said secondary component is selected from the group consisting of a (sub-) table, an image, and text.

8. A method according to claim 7 wherein said adjusting comprises applying a plurality of differing scaling factors to widths of said secondary components.

9. A method according to claim 1 wherein said components are selected from the group consisting of text, images and tables, said tables comprising sub-components selected from the group consisting of text, images and tables.

10. A method according to claim 1 wherein said window comprises a vertical scroll bar and is characterised by the absence of an active horizontal scroll bar.

11. A method according to claim 1 wherein said information comprises data formatted according to a protocol for presentation.

12. A method according to claim 11 wherein said protocol comprises HTML and said information comprises a Web page having elements collectively arranged for presentation on a Web browser application according to a first format, and said adjusting modifies the widths of said elements so that an entire width of Web page is displayed within a width of said window.

13. Apparatus for performing the method of any one of claims 1 to 12.

14. A computer network browser application comprising:

a graphical user interface including a window display area having a predetermined width;

means for retrieving a page of information from a computer network, said information comprising plural components;

means for determining those first ones of said components that contribute to a width of said page;

means for adjusting display sizes of said first components by applying a plurality of differing scaling factors to widths of said first components, so that said width of said

page is adjusted to fall within the width of said window display area while maintaining a layout corresponding to said page;

means for displaying the adjusted information in said display area.

15. An application according to claim 14, wherein said adjusted information as displayed comprises at least a user selectable element forming part of said graphical user interface.

16. An application according to claim 14, wherein said width of at least one of said first components is adjusted by resizing said first component.

17. An application according to claim 14 wherein at least one of said components is a text component, and said adjusting of said display size of said text component comprises wrapping text within said text component to adjust the width of said text component.

18. A computer program element comprising computer program code means to make a computer execute a procedure to present information comprising plural components, said components being collectively arranged for presentation, said computer program element comprising:

code for establishing a presentation window having a width;

code for determining those first ones of said components that contribute to a width of said presentation;

code for adjusting display sizes of said first components by applying a plurality of differing scaling factors to widths of said first components, so that said width of said presentation is adjusted to fall within said width of said presentation window while maintaining a layout corresponding to said presentation; and

code for displaying the adjusted information within said presentation window.

19. A computer readable medium, having a program recorded thereon, where the program is configured to make a computer execute a procedure to present information comprising plural components, said components being collectively arranged for presentation, said procedure comprising the steps of:

establishing a presentation window having a width;

determining those first ones of said components that contribute to a width of said presentation;

adjusting display sizes of said first components by applying a plurality of differing scaling factors to widths of said first components, so that said width of said presentation is adjusted to fall within said width of said presentation window while maintaining a layout corresponding to said presentation; and

displaying the adjusted information within said presentation window.

20. A method of presenting information substantially as described herein with reference to any one of Fig. 3, Fig. 4, Fig. 5B, Fig. 6B, Fig. 7A, Fig. 7B, Figs. 11A and 11B of the drawings.

21. A method of presenting information substantially as described herein with reference to Appendix A.

22. A Web browser system substantially as described herein with reference to any combination of Fig. 3, Fig. 4, Fig. 5B, Fig. 6B, Fig. 7A, Fig. 7B, Fig. 8C, Figs. 10A-10D, of the drawings and Appendix A.

DATED this Twenty First Day of March 2003

Canon Kabushiki Kaisha

Patent Attorneys for the Applicant

SPRUSON & FERGUSON

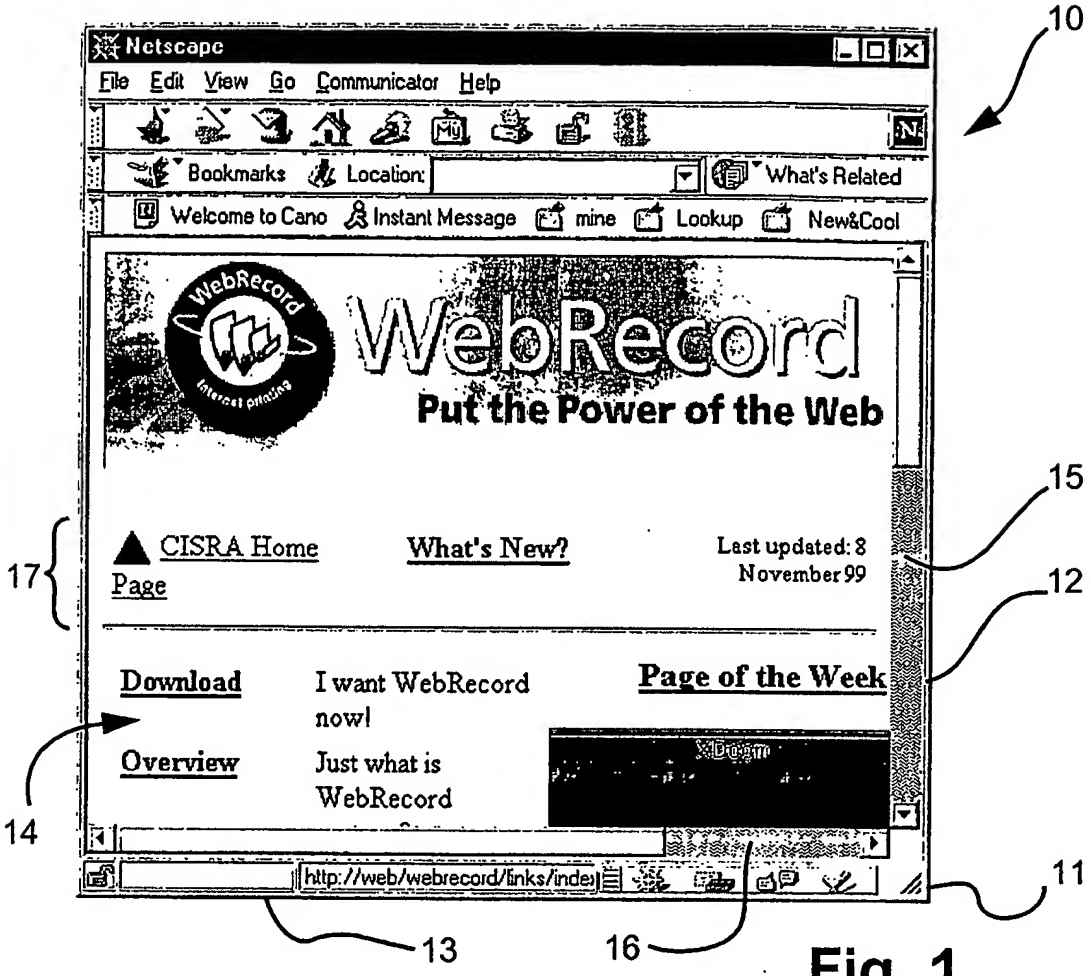

Fig. 1
(Prior Art)

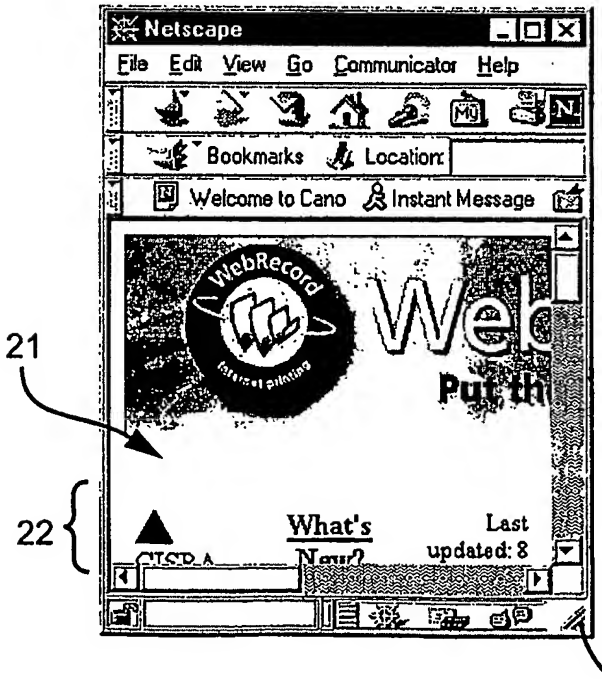

Fig. 2
(Prior Art)

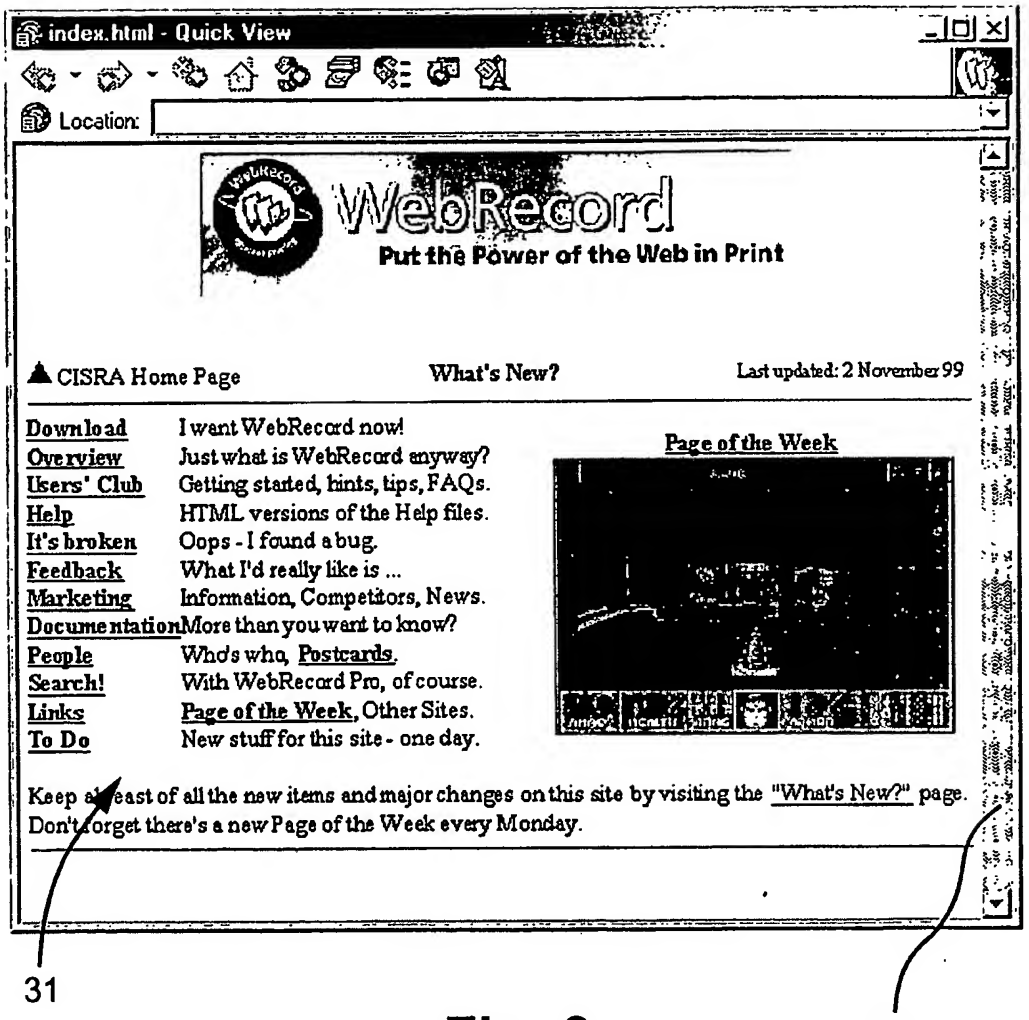
30
index.html - Quick View
Location:
WebRecord
Put the Power of the Web in Print
CISRA Home Page
What's New?
Last updated: 2 November 99
Download I want WebRecord now!
Overview Just what is WebRecord anyway?
Users' Club Getting started, hints, tips, FAQs.
Help HTML versions of the Help files.
It's broken Oops - I found a bug.
Feedback What I'd really like is ...
Marketing Information, Competitors, News.
Documentation More than you want to know?
People Who's who, Postcards.
Search! With WebRecord Pro, of course.
Links Page of the Week, Other Sites.
To Do New stuff for this site - one day.
Page of the Week
Keep abreast of all the new items and major changes on this site by visiting the "What's New?" page.
Don't forget there's a new Page of the Week every Monday.
31
32

Fig. 3

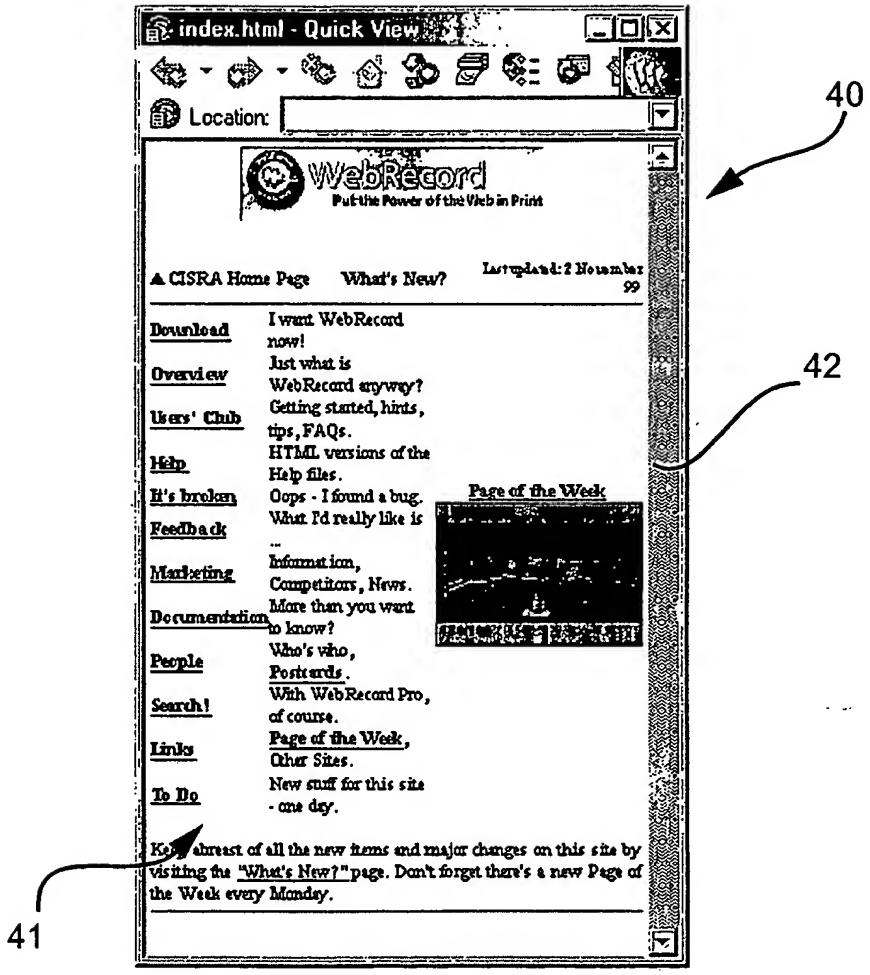
index.html - Quick View
Location:
WebRecord
Put the Power of the Web in Print
▲ CISRA Home Page
What's New?
Last updated: 2 November 99
Download
I want WebRecord now!
Overview
Just what is WebRecord anyway?
Users' Club
Getting started, hints, tips, FAQs.
Help
HTML versions of the Help files.
It's broken
Oops - I found a bug.
Feedback
What I'd really like is ...
Marketing
Information, Competitors, News.
Documentation
More than you want to know?
People
Who's who, Postcards.
Search!
With WebRecord Pro, of course.
Links
Page of the Week, Other Sites.
To Do
New stuff for this site - one day.
Page of the Week
Keep abreast of all the new items and major changes on this site by visiting the "What's New?" page. Don't forget there's a new Page of the Week every Monday.
40
41
42

Fig. 4

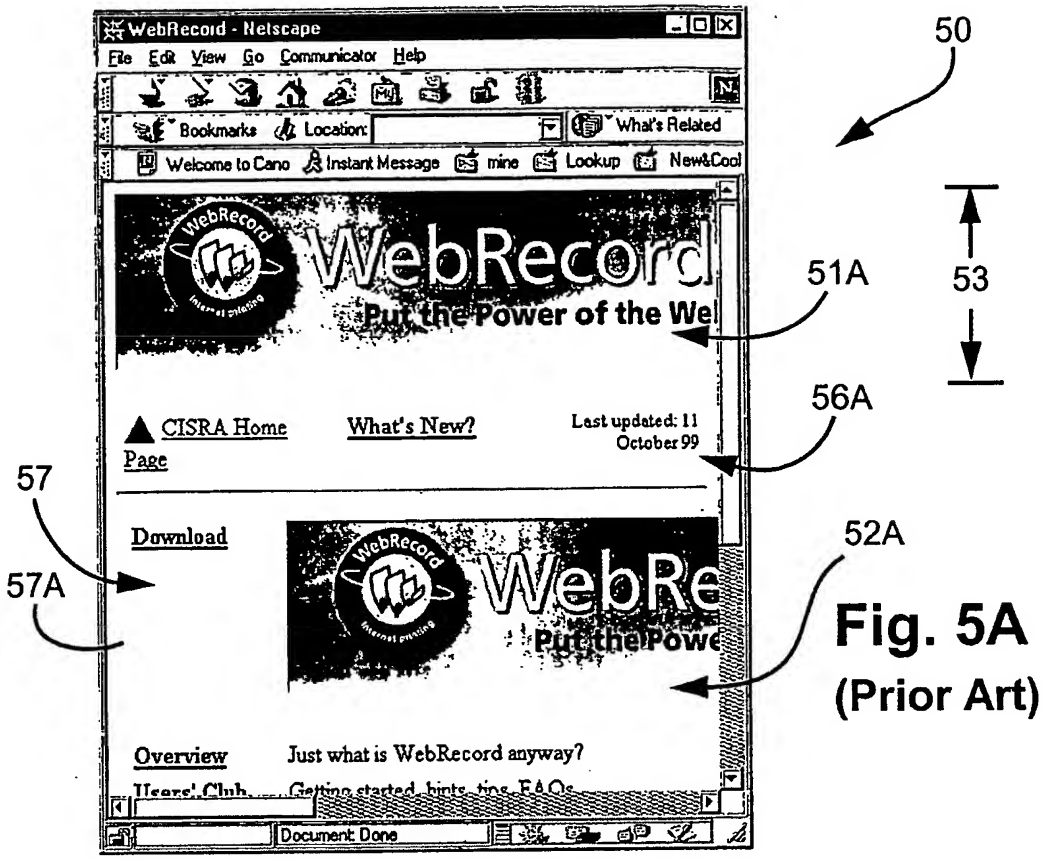

Fig. 5A
(Prior Art)

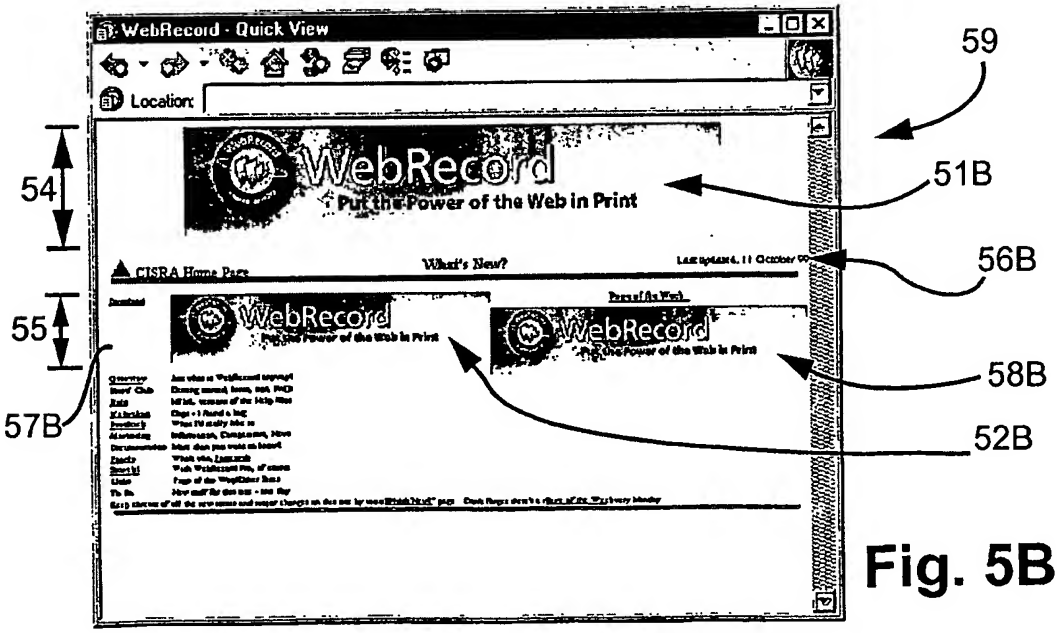

Fig. 5B

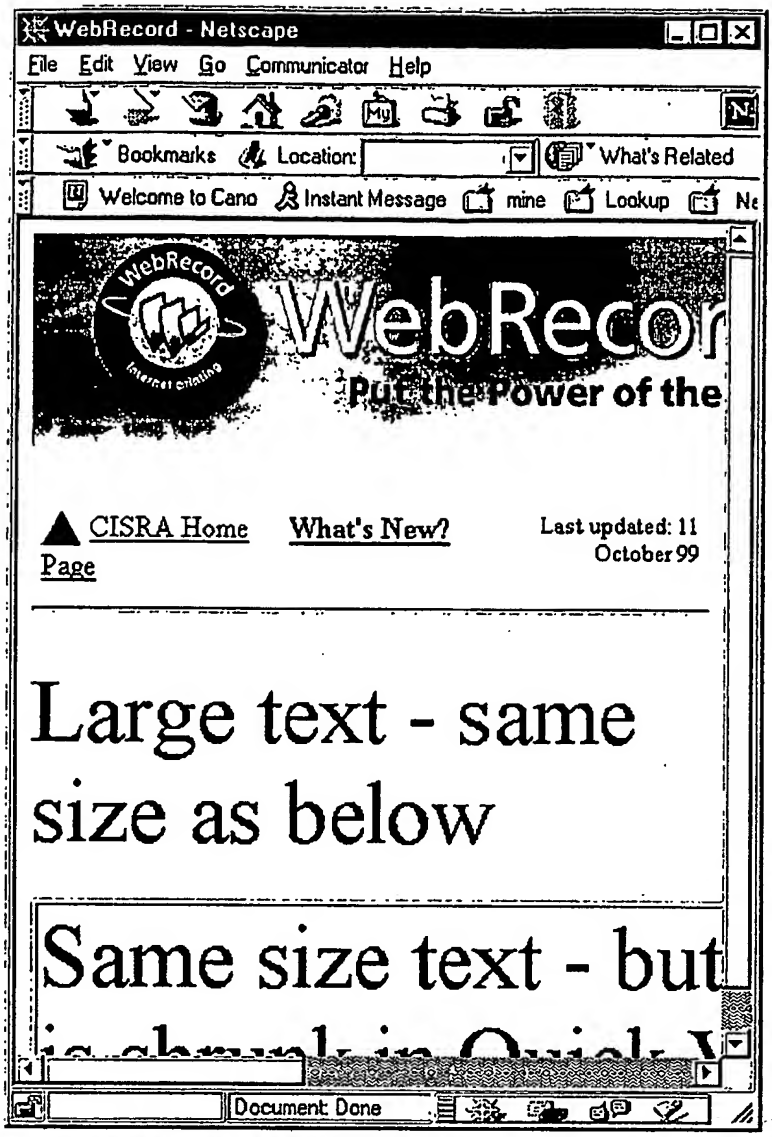

Fig. 6A
(Prior Art)

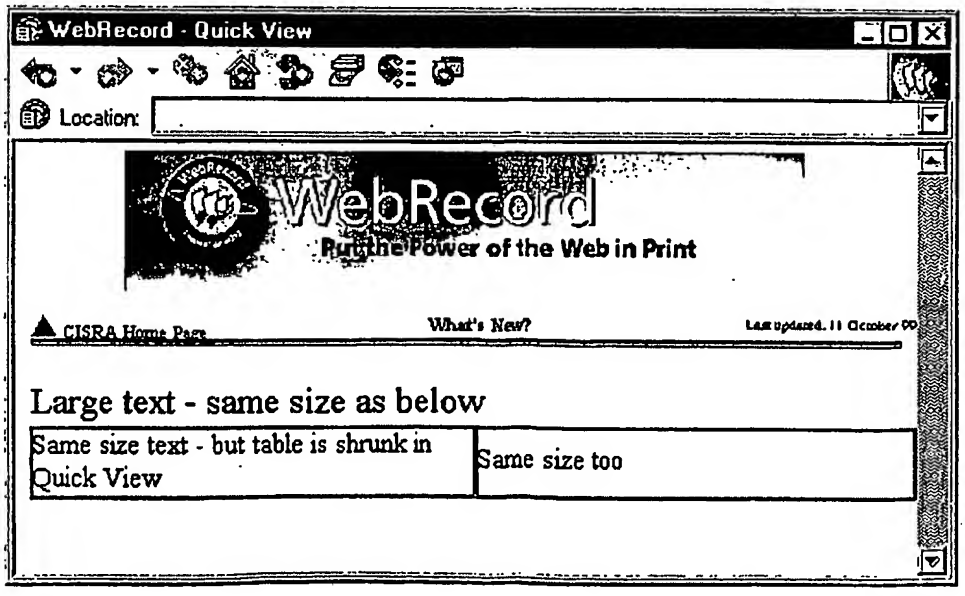

Fig. 6B

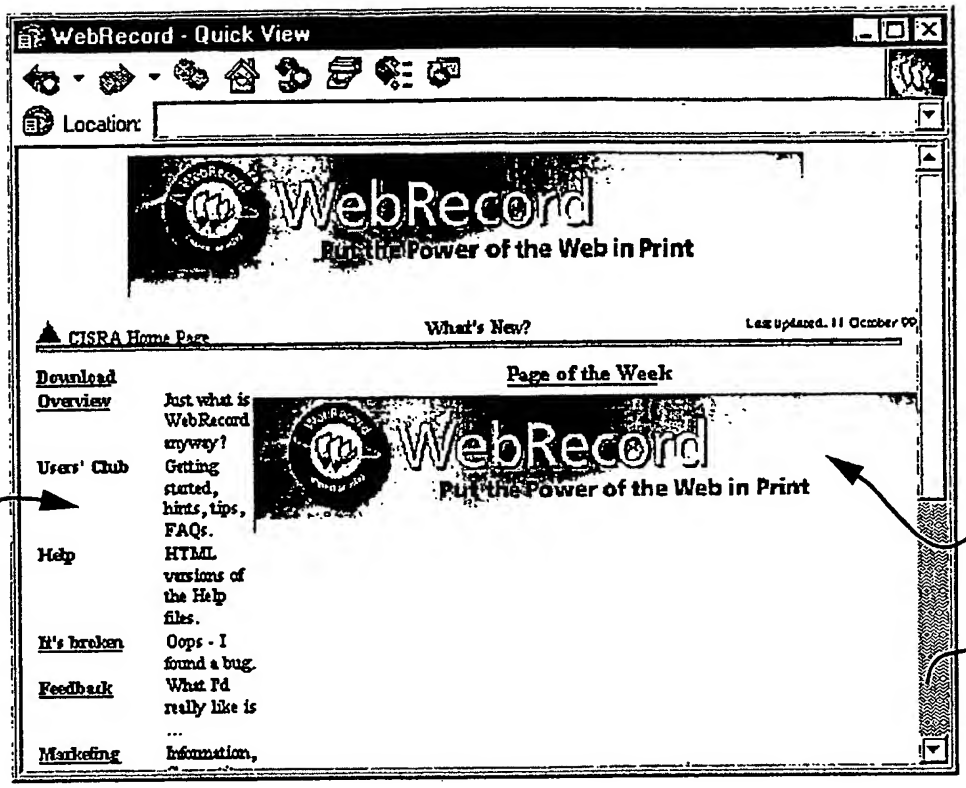
WebRecord - Quick View
Location:
WebRecord
Put the Power of the Web in Print
CISRA Home Page
What's New?
Last updated: 11 October 99
Download
Overview
Just what is WebRecord anyway?
Users' Club
Getting started, hints, tips, FAQs.
Help
HTML versions of the Help files.
It's broken
Oops - I found a bug.
Feedback
What I'd really like is ...
Marketing
Information,
Page of the Week
WebRecord
Put the Power of the Web in Print
70
71
72
73

Fig. 7A

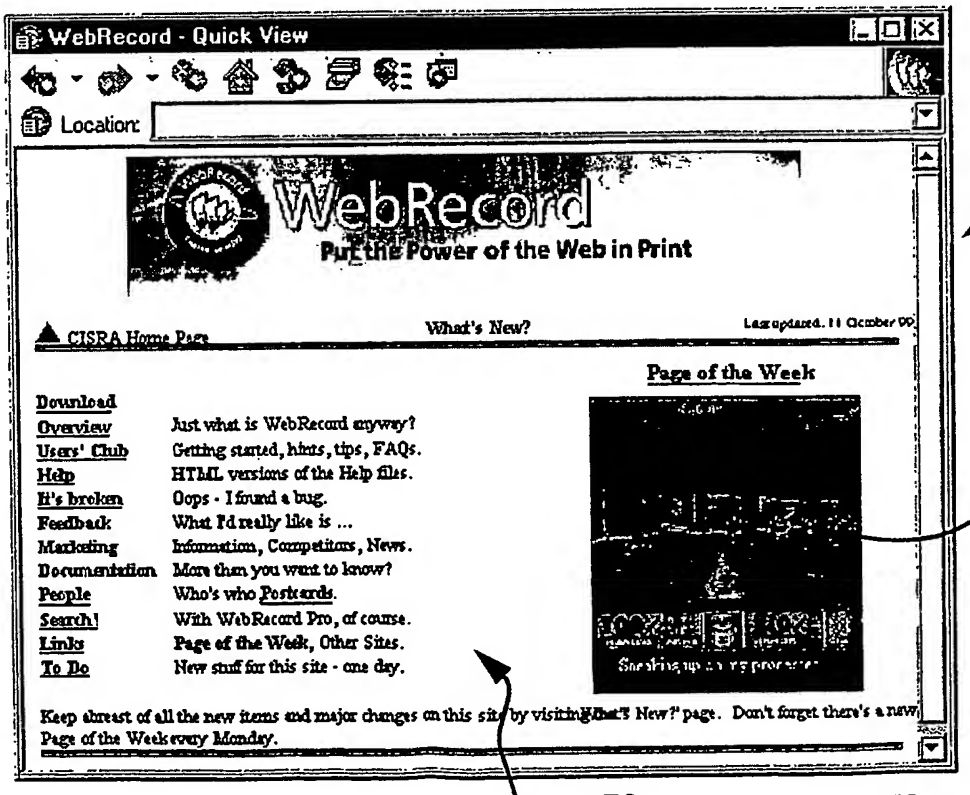
WebRecord - Quick View
Location:
WebRecord
Put the Power of the Web in Print
CISRA Home Page
What's New?
Last updated: 11 October 99
Page of the Week
Download
Overview
Just what is WebRecord anyway?
Users' Club
Getting started, hints, tips, FAQs.
Help
HTML versions of the Help files.
It's broken
Oops - I found a bug.
Feedback
What I'd really like is ...
Marketing
Information, Competitors, News.
Documentation
More than you want to know?
People
Who's who Postcards.
Search!
With WebRecord Pro, of course.
Links
Page of the Week, Other Sites.
To Do
New stuff for this site - one day.
Keep abreast of all the new items and major changes on this site by visiting the "What's New?" page. Don't forget there's a new Page of the Week every Monday.
74
75
76

Fig. 7B

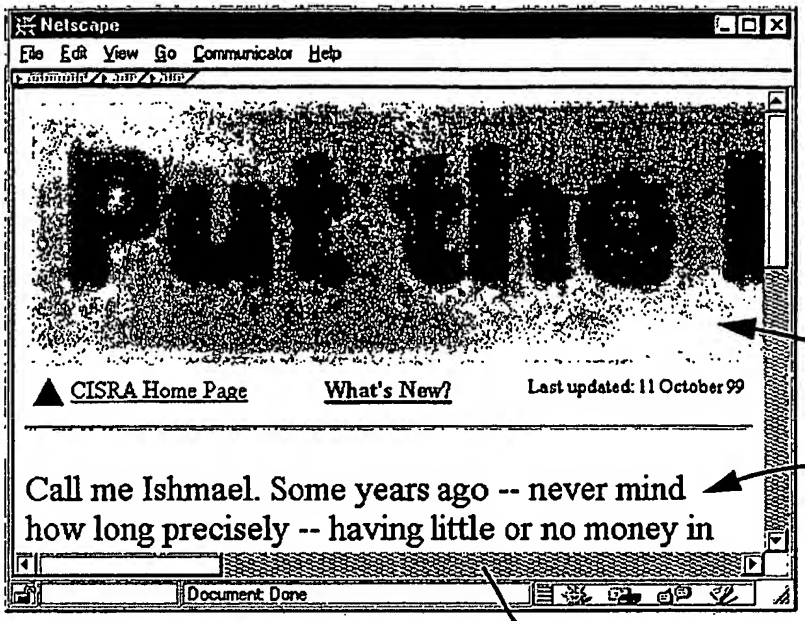

Fig. 8A
(Prior Art)

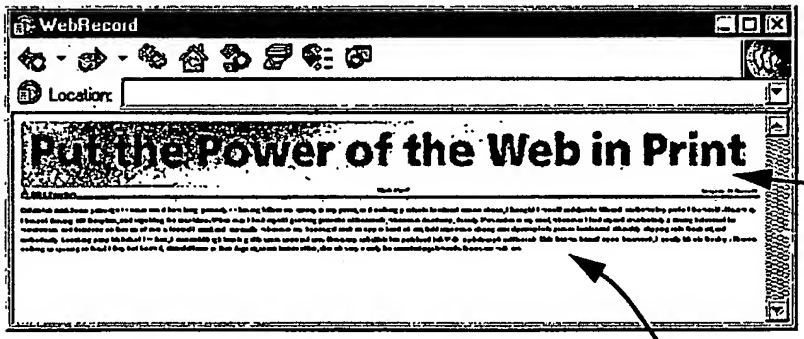

Fig. 8B
(Prior Art)

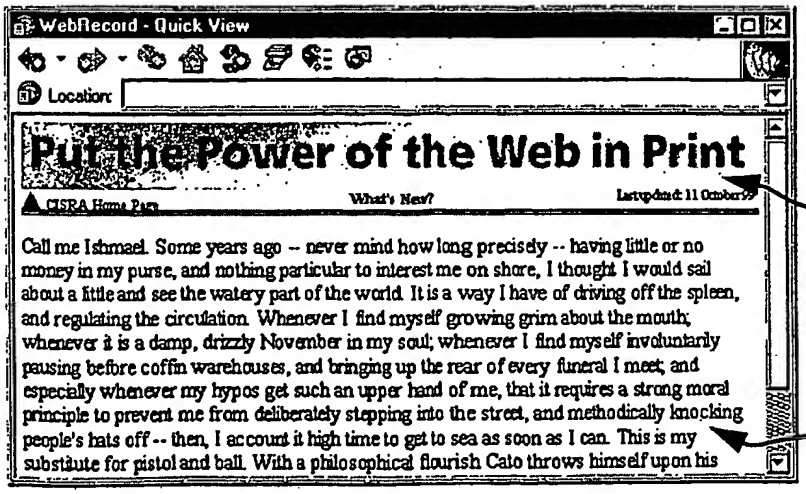

Fig. 8C

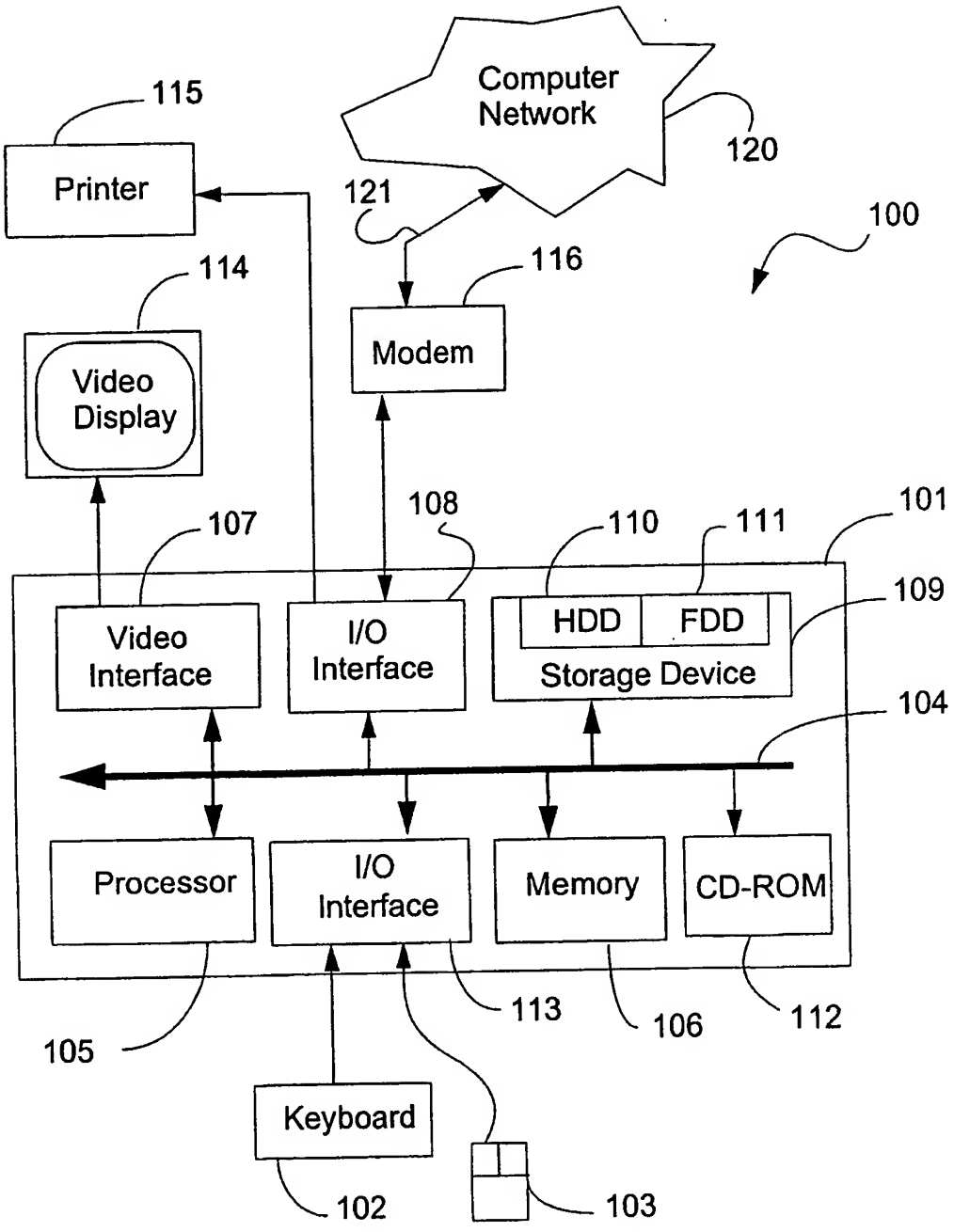
Computer Network
120
115
Printer
121
100
116
114
Modem
Video Display
108
107
110
111
101
109
Video Interface
I/O Interface
HDD
FDD
Storage Device
104
Processor
I/O Interface
Memory
CD-ROM
105
113
106
112
Keyboard
102
103

Fig. 9

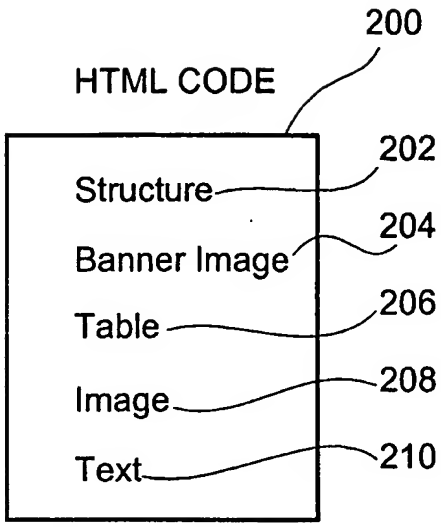

Fig. 10A

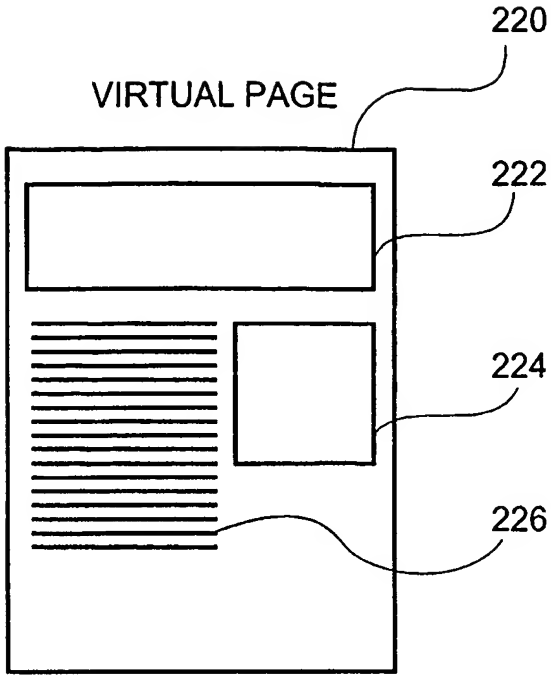

Fig. 10B

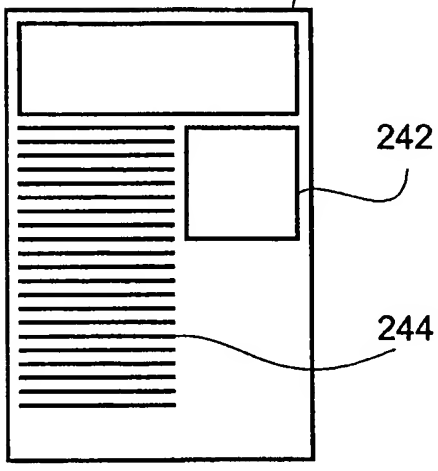

Fig. 10D

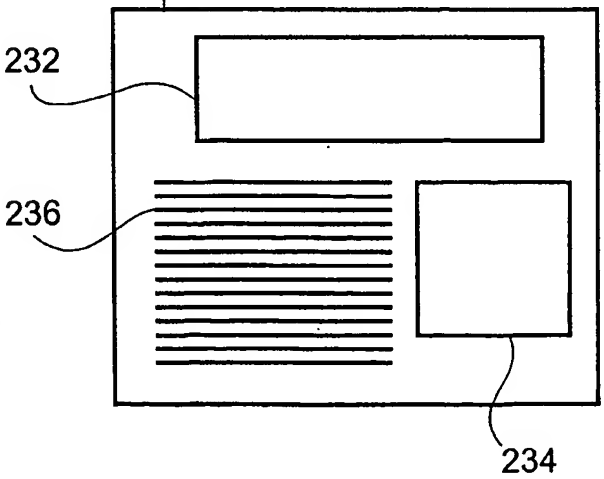

Fig. 10C

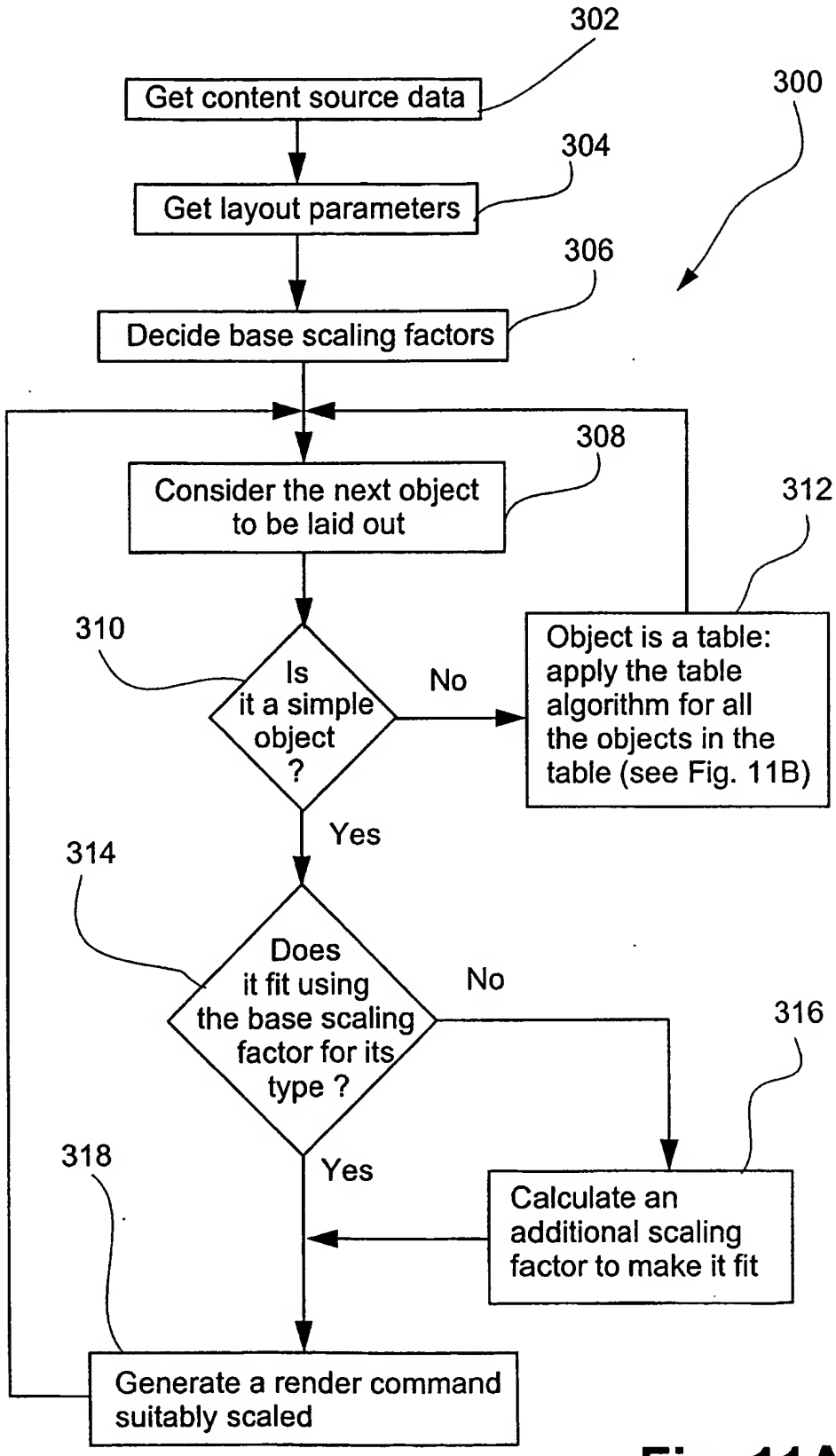
302
Get content source data
300
304
Get layout parameters
306
Decide base scaling factors
308
Consider the next object to be laid out
312
310
Is it a simple object ?
No
Object is a table: apply the table algorithm for all the objects in the table (see Fig. 11B)
Yes
314
Does it fit using the base scaling factor for its type ?
No
316
Yes
318
Calculate an additional scaling factor to make it fit
Generate a render command suitably scaled

Fig. 11A

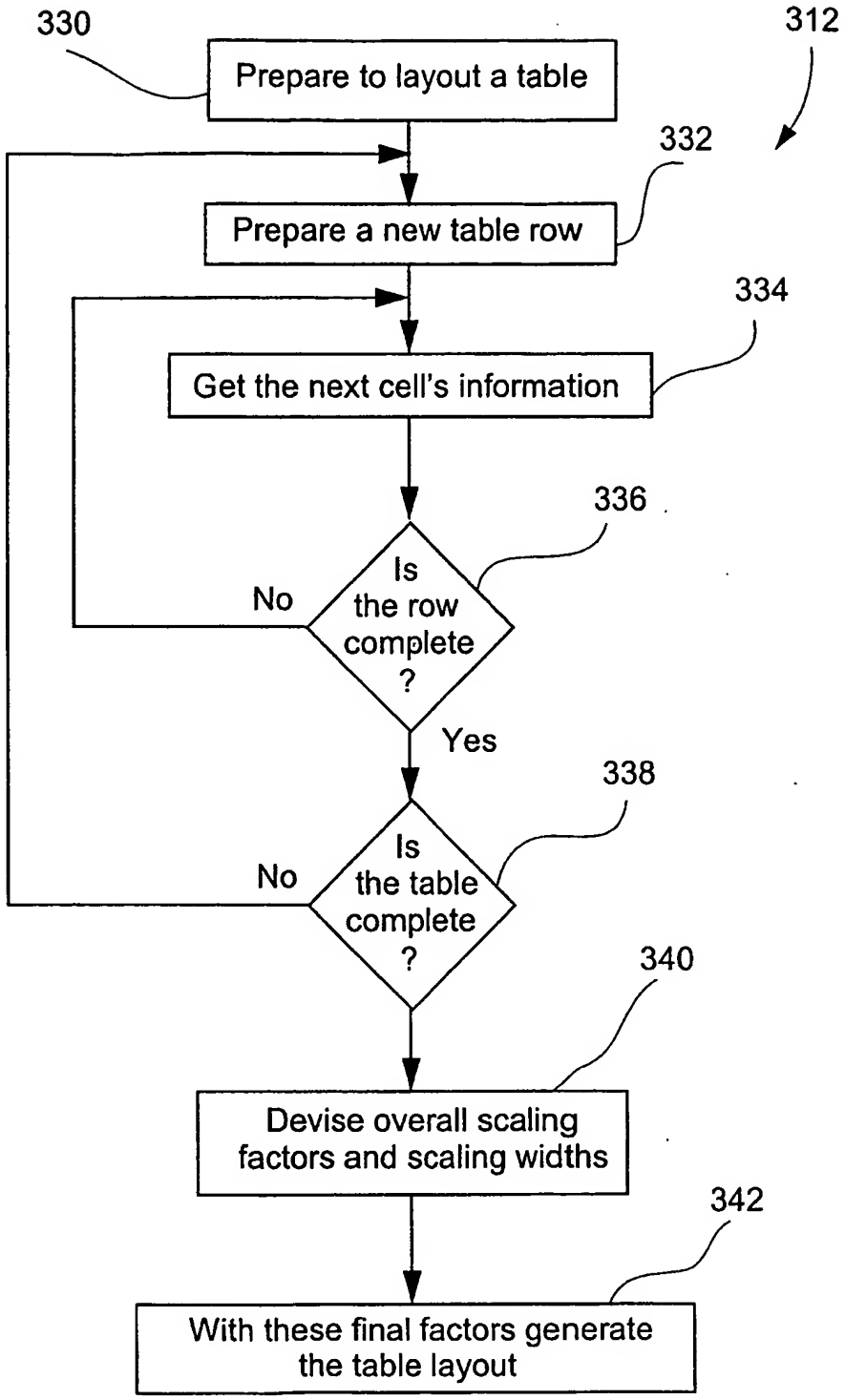
330
Prepare to layout a table
312
332
Prepare a new table row
334
Get the next cell's information
336
Is the row complete ?
No
Yes
338
Is the table complete ?
No
340
Devise overall scaling factors and scaling widths
342
With these final factors generate the table layout

Fig. 11B

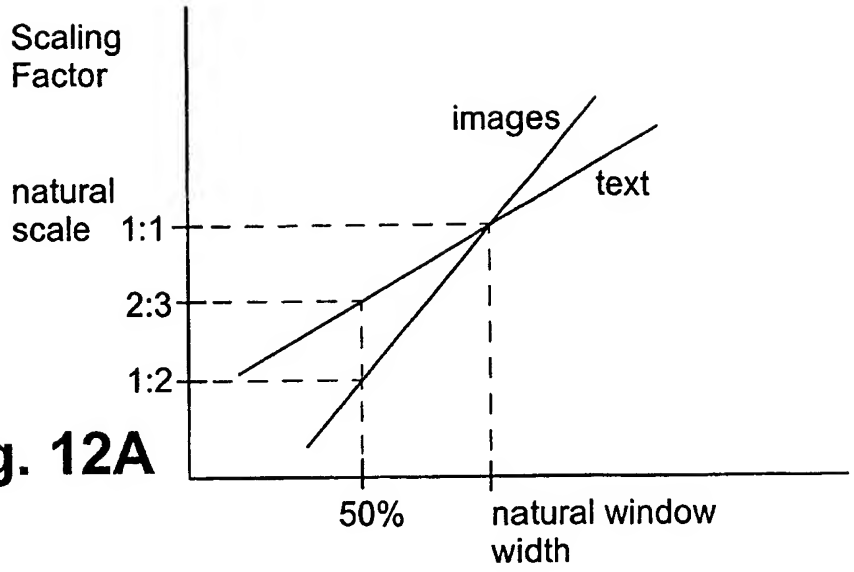

Fig. 12A

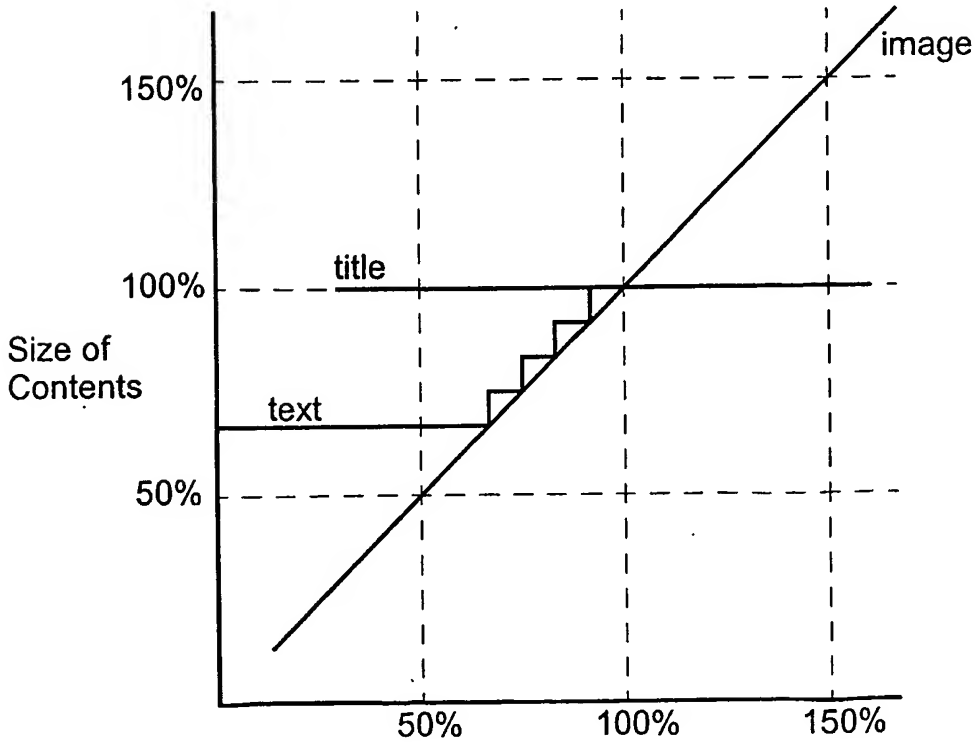

Fig. 12B